User's Manual

WRAP 'N' SHAKE

Systematic exploration of multiple drug binding sites

Version 1.1

www.wnsdock.xyz

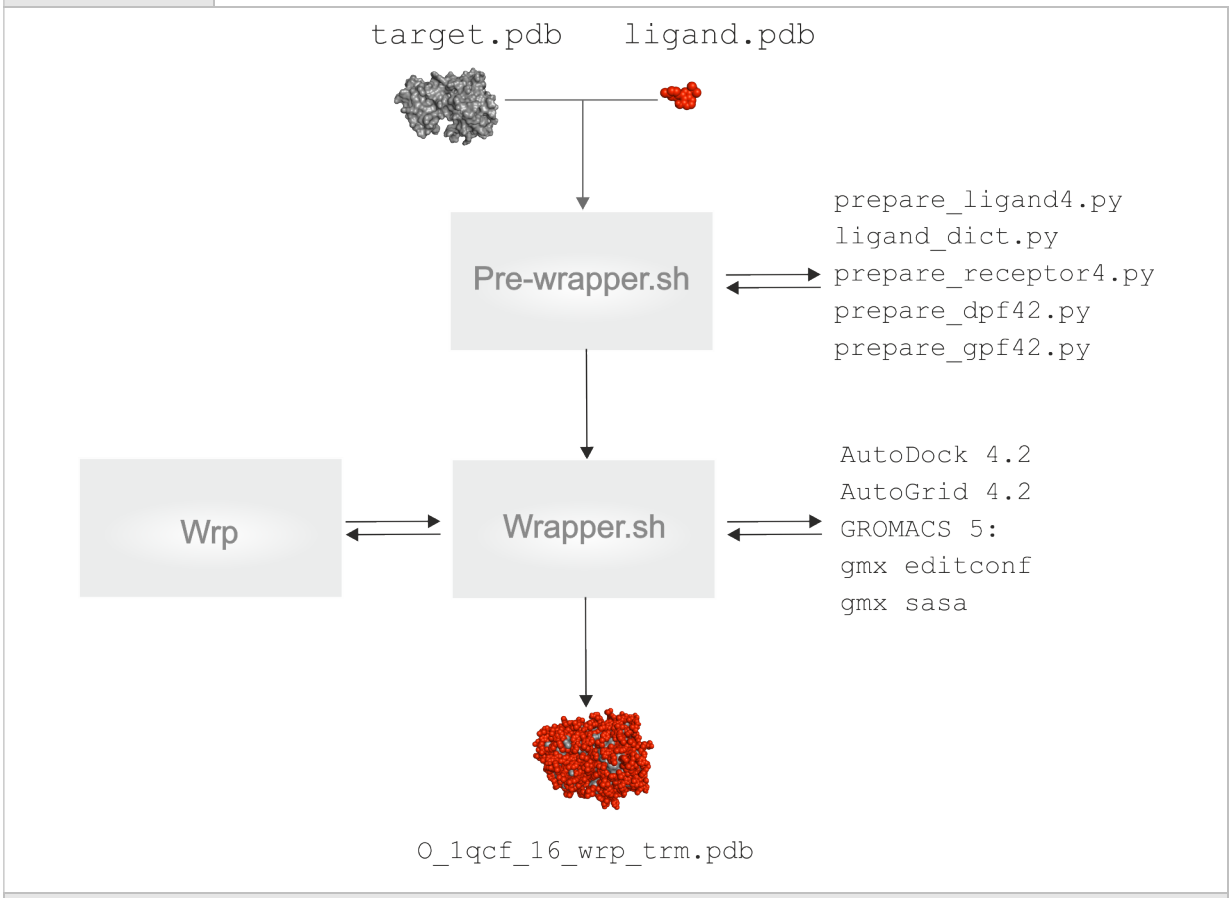
Updated 20-12-2018

Table of Contents

1 Introduction

The Wrap 'n' Shake (WnS) method was designed for systematic exploration of multiple drug binding sites. WnS starts with Wrapper (**Fig. 1**), which systematically covers the entire surface of the target with a monolayer of ligand copies. Wrapper performs several fast blind docking (BD) cycles using AutoDock 4.2.3. In the monolayer, ligand-ligand interactions are minimized, as the ligand copies mostly interact with the target. Wrapper includes two shell scripts (pre-wrapper.sh and wrapper.sh) and a C program (wrp).

Fig. 1



Components of Wrapper (grey boxes) and their connection with external shell scripts and programs

2 Pre-wrapper.sh

Overview

Pre-wrapper.sh is a bash shell script for preparation of AutoDock 4.2 Release 4.2.3 and AutoGrid 4.2 (1) inputs.

Installation

Extract the compressed wns source package (wns.tgz)

```
$ tar -xvf wns.tgz
```

Pre-wrapper.sh can be found in wns/scripts and used under Linux as it is. As necessary inputs, requires the installation of Python scripts of AutoDockTools (2). The list of necessary external python scripts is presented in Table 1.

Input

Energy-minimized structures of the target and ligand molecules are the main inputs. The commonly used Protein Databank (PDB) file format is accepted. Hydrogen atoms are not added to the input PDB files. It is assumed that input structures are complete to have an integer total charge on both molecules (target and ligand). If the user does not have a minimized ligand or target structure, with hydrogens added, hydrogens can be added in the graphical interphase of ADT, or by modifying the pre-wrapper.sh script, by adding "–A hydrogens" to the prepare_receptor4.py and prepare_ligand4.py command lines.

e.g.:

```
$SCRIPTPATH/pythonsh $SCRIPTPATH/prepare_receptor4.py -r $target -o
$target_name.pdbqt –A hydrogens -v -U nphs
```

and

```
$SCRIPTPATH/pythonsh $SCRIPTPATH/prepare_ligand4.py -l $ligand -o
$ligand_name.pdbqt –A hydrogens -v -d $SCRIPTPATH/ligand_dict.py -F
```

Algorithm

The pre-rapper.sh script generates *.pdbqt files and the corresponding parameter files (*.dpf and *.gpf) for each target and ligand pair. Pre-wrapper.sh adds the actual path of the modified AD4_parameters.dat (**Fig. 2**) file as an additional first line to *.dpf and *gpf. This tells the programs of AutoDock 4.2 and AutoGrid 4.2, to use the modified AD4_parameters.dat file instead of the default one normally stored in source code folder. The first row of the parameter files (both the DPF and GPF) is updated to the actual path of the modified AD4_parameters.dat, which is user-defined.

Default:
autodock_parameter_version 4.2
Modified:

parameter_file $USER_DEFINED_PATH/AD4_parameters.dat

Pre-wrapper.sh also adds new entries of excluded atom types LL and YY (commonly marked as X in our original publication (3) to the *.dpf and *.gpf files. This step is performed only once, as the same parameter files can be used in all wrapping cycles later. This step is necessary for generation of the corresponding new grid map files (*LL.map and *YY.map). New lines of atom types LL and YY are inserted after the last line of standard atom type maps.

map 1qcf_target.YY.map
map 1qcf_target.LL.map

Two lines of atom types LL and YY are inserted to the end of AD4_parameters.dat file too (see Fig. 2). This modified file can be also downloaded from our web page (4).

atom par YY 3.60 1E-04 00.0000 0.00000 0.0 0.0 0 0 0 0
atom par LL 3.60 1E-04 00.0000 0.00000 0.0 0.0 0 0 0 0

The use of pre-wrapper.sh is not mandatory for Wrapper. Generation of the input *.pdbqt and parameter files can be performed as detailed above, with the AutoDockTools and the above modifications can be inserted manually. However, to avoid human mistakes, the use of pre-wrapper.sh is recommended especially if multiple target files, or a library of ligand structures are handled.

Usage

Pre-wrapper.sh can be launched by the following command, where target.pdb, and ligand.pdb files and the path of external scripts are obligatory inputs.

```
$pre-wrapper.sh -t target.pdb -l ligand.pdb -p /path_to_files
```

Output

Pdbqt files of the ligand and target molecules, docking and grid parameter files with updated path to the modified AD4_parameters.dat file are produced.

Table 1 Pre-wrapper.sh		
Input files and arguments (mandatory)		
-t	target.pdb	Target structure minimized and prepared for docking
-l	ligand.pdb	Ligand structure minimized and prepared for docking
-p	/path_to_files	Full path of external scripts (below), and the modified AD4_parameters.dat file
Output files		
target.pdbqt		Target file in pdbqt format used as docking input
ligand.pdbqt		Ligand file in pdbqt format used as docking input

`target.dpf`	Docking parameter file
`target.gpf`	Grid parameter file
External scripts	
`prepare_ligand4.py`	External Python scripts are available as part of AutoDockTools.
`ligand_dict.py`	
`prepare_receptor4.py`	
`prepare_dpf42.py`	
`prepare_gpf4.py`	

Fig. 2

```
atom_par H      2.00  0.020   0.0000   0.00051  0.0  0.0  0  -1  -1  3      #Non H-bonding Hydrogen
atom_par HD     2.00  0.020   0.0000   0.00051  0.0  0.0  2  -1  -1  3      #Donor 1 H-bond Hydrogen
atom_par HS     2.00  0.020   0.0000   0.00051  0.0  0.0  1  -1  -1  3      #Donor S Spherical Hydrogen
atom_par C      4.00  0.150  33.5103  -0.00143  0.0  0.0  0  -1  -1  0      #NonH-bonding Aliphatic Carbon
atom_par A      4.00  0.150  33.5103  -0.00052  0.0  0.0  0  -1  -1  0      #Non H-bonding Aromatic Carbon
atom_par N      3.50  0.160  22.4493  -0.00162  0.0  0.0  0  -1  -1  1      #Non H-bonding Nitrogen
atom_par NA     3.50  0.160  22.4493  -0.00162  1.9  5.0  4  -1  -1  1      #Acceptor 1 H-bond Nitrogen
atom_par NS     3.50  0.160  22.4493  -0.00162  1.9  5.0  3  -1  -1  1      #Acceptor S Spherical Nitrogen
atom_par OA     3.20  0.200  17.1573  -0.00251  1.9  5.0  5  -1  -1  2      #Acceptor 2 H-bonds Oxygen
atom_par OS     3.20  0.200  17.1573  -0.00251  1.9  5.0  3  -1  -1  2      #Acceptor S Spherical Oxygen
atom_par F      3.09  0.080  15.4480  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Fluorine
atom_par Mg     1.30  0.875   1.5600  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Magnesium
atom_par MG     1.30  0.875   1.5600  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Magnesium
atom_par P      4.20  0.200  38.7924  -0.00110  0.0  0.0  0  -1  -1  5      #Non H-bonding Phosphorus
atom_par SA     4.00  0.200  33.5103  -0.00214  2.5  1.0  5  -1  -1  6      #Acceptor 2 H-bonds Sulphur
atom_par S      4.00  0.200  33.5103  -0.00214  0.0  0.0  0  -1  -1  6      #Non H-bonding Sulphur
atom_par Cl     4.09  0.276  35.8235  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Chlorine
atom_par CL     4.09  0.276  35.8235  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Chlorine
atom_par Ca     1.98  0.550   2.7700  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Calcium
atom_par CA     1.98  0.550   2.7700  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Calcium
atom_par Mn     1.30  0.875   2.1400  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Manganese
atom_par MN     1.30  0.875   2.1400  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Manganese
atom_par Fe     1.30  0.010   1.8400  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Iron
atom_par FE     1.30  0.010   1.8400  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Iron
atom_par Zn     1.48  0.550   1.7000  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Zinc
atom_par ZN     1.48  0.550   1.7000  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Zinc
atom_par Br     4.33  0.389  42.5661  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Bromine
atom_par BR     4.33  0.389  42.5661  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Bromine
atom_par I      4.72  0.550  55.0585  -0.00110  0.0  0.0  0  -1  -1  4      #Non H-bonding Iodine
atom_par YY     3.60  1E-04  00.0000   0.00000  0.0  0.0  0   0   0  0      #Excluded target atom
atom_par LL     3.60  1E-04  00.0000   0.00000  0.0  0.0  0   0   0  0      #Excluded ligand atom
```

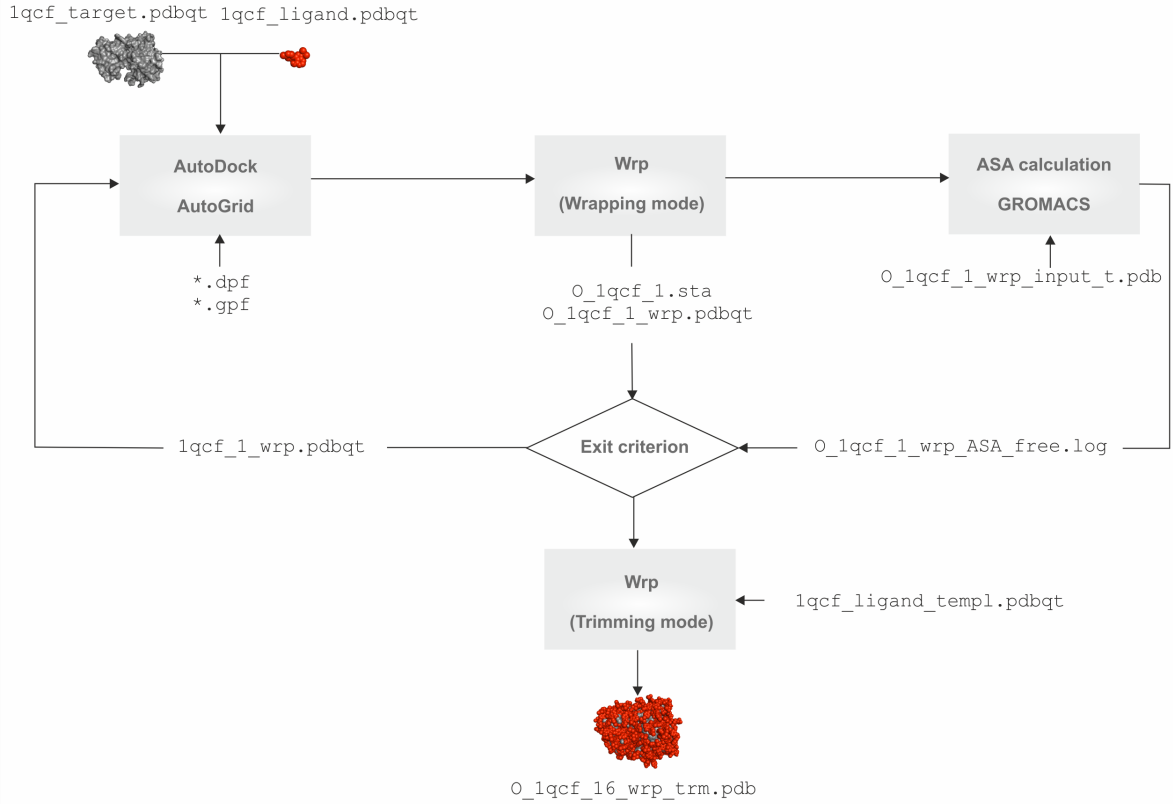
End part of the modified AD4_parameters.dat. The modifications included in the file are highlighted in yellow. Note that atom types LL and YY have the same parameters and commonly mentioned as X in the original publication of WnS (3).

3 Wrapper.sh

Overview

Wrapper.sh (**Fig. 3**) is bash shell script, the director of Wrapper. It performs multiple cycles of blind docking, and automatically creates a monolayer of ligand copies systematically covering the entire surface of the target molecule.

Fig. 3



Wrapper.sh Input and output files names correspond to the example files, presented on web site of WnS (www.wnsdock.xyz). In the presented files, 1qcf stands for the System name, and 16 for the total number of wrapper cycles performed

Installation

Extract the compressed wns source package (wns.tgz)

```
$ tar -xvf wns.tgz
```

Pre-wrapper.sh and wrapper.sh can be found in wns/scripts and are readily usable under the Linux operating system. The source code of wrp can be compiled and installed into $HOME/bin using the following commands.

```
$ cd wns/wrp/src
$ make
$ make install
```

Wrapper.sh can be found in wns/scripts and used under Linux as it is. It requires AutoDock 4.2 Release 4.2.3 and AutoGrid 4.2 programs of the AutoDock 4.2 package. Editconf and sasa programs of the GROMACS package are also required for calculation of free surface of the target. Wrapper works in symbiosis with wrp, included in the present package and detailed in the next Section.
In order to be able to produce all the necessary map files in case of multiple target files, or a library, the source code of AutoDock 4.2 have to be modified. The original source code limits the number grid maps to 14 atom types. To produce grid maps for all atom types, line 93 of autocomm.h file must be modified to a large number (e.g. 34) before installation of AutoDock 4.2 as it follows:

Original:

```
#define MAX_ATOM_TYPES (14 - NUM_NON_VDW_MAPS)     /* Maximum number of atom
types set to keep MAX_MAPS a power of 2 */
```

Modified:

```
#define MAX_ATOM_TYPES (34 - NUM_NON_VDW_MAPS)     /* Maximum number of atom
types set to keep MAX_MAPS a power of 2 */
```

Input

Wrapper.sh requires the files generated with Pre-wrapper.sh. An additional input, the ligand template file (*ligand_templ.pdbqt) is required for post-processing the wrapped target. The template file is used in the trimming mode of Wrapper. The template file is mainly required for two reasons:

- after ligand minimization, all atom names are renamed by MOPAC;
- during Wrapper, all ligand copies are renamed as “LIG” by the wrp program.

Thus, the ligand template file is used for re-numbering and re-naming the ligand atoms and residues name after Wrapper. This ensures an exact match of the ligand atom names, and ligand residue name with the molecular dynamics topology, which is required if the user merges the target ligand complex to use in a Shaker step. The atoms of the template file must have exactly the same order and number of heavy atoms as the input ligand.pdbqt file.
A ligand template file can be prepared by following the same steps as for docked ligand pdbqt, except minimization. Note, that all hydrogen atoms must be added, and the MOPAC energy minimization step is not required. After adding all hydrogen atoms, the PDB template file can be converted to a PDBQT file, using the command line of the python script below, or the graphical interface of ADT program.

$pythonsh $PATH_TO/prepare_ligand4.py -l ligand_templ.pdb -o ligand_templ.pdbqt -v -d $PATH_TO/ligand_dict.py –F

In this way, the same number and order of atoms is obtained in the template file as in the input PDBQT of the ligand. Wrapper.sh also produces log files containing reports on finished cycles with interaction energy and accessible surface values.

In wrapper.sh a maximal cycle number is set by default to 10000, however the user is advised to introduce a smaller, and more realistic maximal cycle number (e.g. 30), to avoid any computational resource waste. The wrapper stage will discontinue if the maximal cycle number, or one of the exit criteria is reached.

Algorithm

Wrapper.sh calls external programs AutoDock 4.2, AutoGrid 4.2 (1), GROMACS 5.0.2 (5) and wrp, an open source C program developed specially for WnS (see Section 4 for a detailed description of wrp). The main steps of the algorithm are described here.

1) A function named cycle is created, to perform the following steps in multiple docking cycles. A docking cycle starts with creating the map files for each atom type specified in the grid parameter file (*.gpf), using AutoGrid 4.2. In the next step, docking is performed by AutoDock 4.2, using the docking parameters specified in the docking parameter file (*.dpf). Besides the docking and grid parameter files, reading the AD4_parameters.dat file form the *.dpf and *.gpf is also necessary in this step as the parameters of the new atom types LL and YY are defined in AD4_parameters.dat as described in Section 2. Function cycle has three arguments: the number of the current cycle, the name of the current target, and the number of the next cycle.

2) Calling program wrp. The docked conformations from the *.dlg file produced by AutoDock 4.2 are clustered and ranked, based on their interaction energy (E_{inter}) values with the target. E_{inter} is the AutoDock 4.2 free energy of binding, without the intramolecular energy terms. The *wrp.pdbqt file produced by Wrp is renamed from O_1qcf_1_wrp.pdbqt into 1QCF_1_wrp.pdbqt, and this will be used as input target file for the next wrapper cycle, if the two exit criterions are not met. Wrp also provides a O_1qcf_1_wrp_input_t.pdb file for the next step.

3) GROMACS programs editconf and sasa are called for calculation of accessible surface area (ASA) using the O_1qcf_1_wrp_input_t.pdb file. The output of the GROMACS programs is shortened to a log file (O_1qcf_1_wrp_ASA_free.log), containing the free target ASA not covered by ligands. Detailed instructions for installation of editconf and sasa programs can be found on the GROMACS web site (http://manual.gromacs.org/current). The editconf command transforms the input pdb file intro gromacs .gro file, and the sasa program performs the calculations. GROMACS sasa calculates the ASA for the entire target-ligand complex, but wrapper.sh will eliminate the surface calculated for the ligand, by deleting rows, with residue name “LIG” from the total_atomarea_lig.xvg file obtained from GROMACS. Wrapper.sh also produces a log file containing the free target surface are not covered by ligand copies.

4) The output file resulted from wrp (O_1qcf_1_wrp.pdbqt) contains the new atom types assigned to the ligand (LL) and target (YY). This file is used as input for the next docking cycle if the two exit criterions are not met. The exit criterions (ASA ≤ 1 % or E_{inter} ≥ 0 kcal/mol) are

checked in each wrapper cycle. The latter is checked in O_1qcf_1.sta file (Table 3), where the E_{inter} of each cluster representative is recorded. The ASA is read from the O_1qcf_1_wrp_ASA_free.log file. If one of the exit criterion is met, then function cycle function will terminate, and a trimming step is started. In the Trimming mode of wrp, ligands that have the distance from the target greater than 3.5 Å, are eliminated.

5) In the folder where the last cycle was performed, and one of the exit criterion was met, track keeping file *surface_exit.log or *energy_exit.log file is generated.

6) The Wrapping process can be restarted, if previous discontinuation occurred prematurely, without reaching one of the exit criterions. In case a restart is necessary, the usage of the wrapper.sh will not change. Wrapper.sh will restart the process from the last complete wrapper cycle.
e.g. If in cycle number 3 O_1qcf_3_wrp.pdbqt is present, then Wrapper.sh will continue from cycle number 4.

Usage

Wrapper.sh can be launched by the following command, where each argument described in Table 2 is an obligatory input.

```
$wrapper.sh -n Maximal count of cycles  -t target.pdbqt -l ligand.pdbqt -b
/home/user/bin -p /home/user/bin -r ligand_templ.pdbqt > wrapper.log
```

Output

In wrp programs Wrapper mode, the target structure wrapped in a monolayer of ligand copies is the final outcome of Wrapper.sh. The structure is stored in a *.pdbqt file as generated by program wrp (see Chapter 4). In the wrp programs Trimming mode, a pdb file is generated (*trm.pdb), which can be readily used in the Shaker stage. Both of these modules are implemented in the wrapper.sh, which generates a standard output, saved as a wrapper.log, which summarizes the current status of Wrapper (**Fig. 4**).

Fig. 4

```
                    WRAPPER Version 1.1, 19.12.2017
Cycle number = 20
Target name = 3PTB_rec.pdbqt
Ligand name = 3PTB_lig.pdbqt
Parameter path = /home/moni/bin
Binary path = /home/moni/bin
Template = BEN_ref.pdbqt
******
****** Wrapper process continues with cycle number: 1!
****** EXIT Condition was not yet reached
****** Started AutoGrid
****** Started AutoDock
****** Surface and energy evaluations
Lowest energy after cycle number 1:
-5.64
******
****** Wrapper process continues with cycle number: 2!
****** EXIT Condition was not yet reached
****** Started AutoGrid
****** Started AutoDock
```

```
****** Surface and energy evaluations
Available free protein surface after cycle number 2:
99.8953
Lowest energy after cycle number 2:
-4.25
******
****** Wrapper process continues with cycle number: 3!
****** EXIT Condition was not yet reached
****** Started AutoGrid
****** Started AutoDock
****** Surface and energy evaluations
Available free protein surface after cycle number 3:
99.8953
Lowest energy after cycle number 3:
-3.44
******
                                                    ….
******
****** Wrapper process continues with cycle number: 8!
****** EXIT Condition was not yet reached
****** Started AutoGrid
****** Started AutoDock
****** Surface and energy evaluations
Available free protein surface after cycle number 8:
96.4815
Lowest energy after cycle number 8:
0.53
******
****** Wrapper process is soon to be finished!
****** ENERGY EXIT Condition was reached at the end of Cycle number: 8
******
Cycle  Energy         ASA
Nr.    (kcal/mol)          (%)
1       -5.64         100.0000
2       -4.25         99.8953
3       -3.44         99.8953
4       -2.80         99.5019
5       -3.27         98.4401
6       -2.06         97.5613
7       -0.68         96.4815
8       0.53          96.4815
```

An example of wrapper.log standard output (3).

Table 2 Wrapper.sh		
Input files and arguments		
-n	Maximal count of cycles	By default, 10000 of maximal cycles is set. To avoid any malfunction of the script, is it is expected from the user to input a smaller, realistic maximal cycle number. In general, n=30 should be enough to cover a target surface with ligand copies.
-t	target.pdbqt	Target pdbqt file
-l	ligand.pdbqt	Ligand pdbqt file
-p	home/user/bin	Path to binary files
-b	home/user/bin	Path to parameter files
-r	ligand_templ.pdbqt	Ligand template file, is used for re-numbering and re-naming the ligand atoms and residue name after docking. Residue numbering in the template file must start with a single digit number (e.g. 1 is advisable). Otherwise, the residue number in the target-ligand$_N$ complex, might add up, exceeding the 3-digit number, which is not permitted by the strict format of the pdb file.

Output files		
`target_1.dlg`		Docking log file
`target_1_ASA_free.log`		Shortened ASA log file, containing the free accessible protein surface area
`surface_exit.log` or `energy_exit.log`		In the exiting cycle directory, one of the two (surface_exit.log or energy_exit.log) files will be generated by wrapper.sh
`Standard output (e.g. wrapper.log)`		The exiting cycle number can be found in the standard output, written out by wrapper.sh (see wrapper.log in Fig. 4)
External programs		
`AuroGrid 4.2`		Calculation of grid maps. (Maximum number of grid points should be set large enough for blind docking on the entire surface of the target. In many cases, 300 grid points are enough for blind docking on common targets.)
`AutoDock 4.2`		The docking engine
`Wrp`		See Section 4
`GROMACS: gmx editconf and gmx sasa`		The surface area of the input target is calculated for each cycle, using GROMACS packages.

Table 3. STA file	
Column name	**Description of the data within the column**
#AD_{Cycle}	The current Wrapper Cycle
#Rank	Number of Ranks of the cluster representatives, ordered by the $E_{intermin}$
$E_{intermin}$	The intermolecular interaction energy of the cluster representative structure, as calculated by the AutoDock 4.2 scoring function.
dG_{min}	The interaction energy of the cluster representative structure, as calculated by the AutoDock 4.2 scoring function. This type of energy contains both the inter and intramolecular ineractions.
EI	Efficiency index, which is calculated from the $E_{intermin}$, divided by the heavy atoms
Pop	Population count of the specific cluster
#min	This number indicates which ligand copy was selected as cluster representative, out of the 100 docked ligand copies
$RMSD_{med}$	RMSD median of the ligand conformations within the cluster
$RMSD_{mad}$	RMSD mean absolute deviation of the ligand conformations within the cluster
$E_{intermed}$	$E_{intermed}$ median of the ligand conformations within the cluster
$E_{intermad}$	$E_{intermad}$ mean absolute deviation of the ligand conformations within the cluster
$RMSD_{mean}$	RMSD mean of the ligand conformations within the cluster
$RMSD_{sd}$	RMSD standard deviation of the ligand conformations within the cluster
$E_{intermean}$	$E_{intermean}$ mean of the ligand conformations within the cluster
$E_{intersd}$	$E_{intersd}$ standard deviation of the ligand conformations within the cluster

4 Wrp

Overview

Wrp is an open source C program and serves as the background engine of module Wrapper. It performs clustering, ranking, and assignation of excluded atoms in Wrapper. Wrp also performs a trimming step. Repeated use of Wrp in Wrapper provides the target structure systematically wrapped in a monolayer of ligand copies.

Input

Wrp is called by Wrapper.sh. Depending on the running mode, different input files are required. The filetypes and their main purpose is detailed in Table 4. In wrapping mode, the docking log (*.dlg) is the main input as resulted from AutoDock 4.2 in a wrapping cycle. In the trimming mode, the wrapped target (*wrp.pdbqt) and the template file (*templ.pdbqt), serves as an input.

Algorithm

Wrapping mode. Wrp has various functions in wrapping mode as listed here.

1) Wrp performs clustering and ranking of the 100 docked ligand conformations listed in the in a log file (1qcf_1.dlg for the first cycle) by AutoDock 4.2. The log file is evaluated by the wrp program, which first ranks and clusters the docked ligand conformations. Docked ligand conformations of the DLG file are clustered and ranked based on their interaction energy (E_{inter}, the 1st energy component of estimated free energy of binding in the DLG file) values with the target, and the closest distance between each heavy atom of the ligand copies (d_{min}). In the initial clustering phase, wrp (wrapper mode) sorts the 100 docked ligand conformations according to $E_{inter.}$ Ligand conformation of the lowest E_{inter} from among the 100 docked ligand copies is selected as the representative of Cluster 1. Ligand conformation of the 2nd lowest E_{inter} is selected as a representative of a new Cluster 2 if d_{min}>drnk, where drnk is a ranking tolerance, a measure of separation of clusters from each-other. If $d_{min}\leq$drnk, then ligand conformation of the 2nd lowest E_{inter} is placed into Cluster 1. In this way, all 100 ligand conformations are clustered and the representatives are evenly spread over the target surface without clashing each-other. In our protocol, drnk was set to 2 Å, which is approximately a covalent bond distance (1.5 Å) plus a 0.5 Å added. The results of clustering are summarized in .sta file type (O_1qcf_1_wrp.sta) after each wrapper cycle.

2) Wrp in wrapper mode assigns the new atom type (YY, LL) of the above-mentioned excluded atoms in the target file (YY), and the docked ligand copies (cluster representatives LL). Excluded atoms are assigned using a target-ligand interface tolerance (difc) and an assignation tolerance (dsgn). Both tolerance values were set to 3.5 Å in the default settings. Merging of the modified target and ligand copies results in a target-ligand complex O_1qcf_1_wrp.pdbqt file. This file is moved from the working directory of the current cycle into the directory of the next cycle, and used as target input for the AutoGrid 4.2 and AutoDock 4.2 if none of exit criteria described below are achieved. After each cycle, the free (unliganded) Accessible Surface Area (ASA) is calculated by external GROMACS program sasa, as described in Section 3 (Msroll in the 1.0 version). Wrapping ends if ASA ≤ 1 % or the interaction energy E_{inter} value of any cluster representative in the cycle is ≥ 0 kcal/mol. Otherwise, the resulted PDBQT file is forwarded to the next cycle. ASA and E_{inter} evaluations

are calculated for each wrapper cycle and stored in two separate files (O_1qcf_1_surface_percentage.log and O_1qcf_1_lowest_energy.log). These files are generated in the working directory of each cycle and moved to "stats" folder where statistical evaluation of Wrapper takes place.

Trimming mode. After the last wrapping cycle a trimming mode of wrp is involved to remove ligand copies positioned farther from the target surface than the trimming tolerance (dmax). This is necessary, as some ligand copies may dock to distant regions of the docking box depending on the actual target. The trimming step also performs formal post-processing of the *.pdbqt file resulted after wrapping, using a template file (1qcf_ligand_templ.pdbqt). The resulted *trm.pdb file has all atoms re-named according to the standards of PDB file format allowing the use of this file of the molecular dynamics steps of a Shaker process.

Usage

By default, wrp is used with silent verbosity in the wrapping mode by wrapper.sh. An example command line is presented below.

```
$ wrp -f target_1_wrp.dlg -p ligand_templ.pdbqt -t target_1_wrp.pdbqt -c cycle nr. -difc 3.500 -drnk 2.000 -dsgn 3.500 -m wrapping -v silent
```

Trimming mode of wrp is used only once after the wrapping process, and an example command line is presented below.

```
$ wrp -f 1qcf_1_wrp.trm.pdbqt -p 1qcf_templ.pdbqt -dmax 3.500 -m trimming -v silent
```

Output

In wrapping mode the outcome is the target structure (partially) covered by ligand copies in a *wrp.pdbqt file. Additionally to the *wrp.pdbqt file, a statistics file (*.sta) is also generated, that contains the E_{inter} of the cluster representatives. After the trimming mode, the *trm.pdb file is generated, which contains the target structure wrapped in a monolayer of N ligand copies and can be readily used in Shaker method.

Table 4 Wrp		
Input files (silent or diagnostic)		
-f	target_1_wrp.dlg	Docking log file, resulted from AutoDock 4.2, containing 100 docked ligand copies
-f	target_1_wrp.pdbqt	Target with ligand copies in trimming mode
-p	ligand_templ.pdbqt	Template ligand, see Table 2
-r	*ref.pdb (optional)	Reference pdb file, used for RMSD calculation (not used in the default wrapper.sh, but can be added by the user) For correct RMSD calculation, the order of the atoms in the reference file, must match exactly the order of the atoms in the docked ligand file.
-t	target_1_wrp.pdbqt	Target (with ligand copies) in wrapping mode
Parameters and defaults		
-c	1	Serial number of wrapping cycle

-difc	3.500	Interface tolerance (Å)
-dmax	3.500	Trimming tolerance (Å)
-drnk	2.000	Ranking tolerance (Å)
-dsgn	3.500	Assignation tolerance (Å)
-m	Wrapping	Program mode, wrapping/trimming
-v	silent	Verbosity, silent/diagnostic
Output files		
Silent		
O_target_1.log		Log file, stating the used parameters
O_taget_1.sta		Statistics file
O_target_1_wrp.pdbqt		The complexed target-ligand file. This is the file that will be used in the following wrp cycle.
O_target_1_input_t.pdb		Target or target – ligand complex file, that was used as input for the given wrapper cycle. This is the file that is used as input for ASA calculations by GROMACS.
Diagnostic		
O_target_1_wrp.pdb		Target-ligand complex file, that will be used in the next wrapper cycle, in pdb format
O_target_1_rank_1.pdb … O_target_1_rank_CC.pdb		Cluster representative, complexed with target residues that were calculated within the Interface tolerance. (CC stands for cluster count).

5 References

1. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, et al. AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility. J Comput Chem. 2009;28(1):73–86.
2. AutoDock Tools 1.5.7 [Internet]. 2009 [cited 2018 Sep 28]. Available from: http://mgltools.scripps.edu/downloads
3. Bálint M, Jeszenői N, Horváth I, van der Spoel D, Hetényi C. Systematic exploration of multiple drug binding sites. J Cheminform [Internet]. Springer International Publishing; 2017;9(1):65. Available from: https://jcheminf.springeropen.com/articles/10.1186/s13321-017-0255-6
4. Wrap'n'Shake [Internet]. 2017 [cited 2018 Oct 1]. Available from: http://www.wnsdock.xyz/
5. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, et al. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. 2015;1–2:19–25.

6 Disclaimer

WRAP 'N' SHAKE (REFERRED TO AS WNS IN THE FORTHCOMING TEXT) IS A FREE SOFTWARE: YOU CAN REDISTRIBUTE IT AND/OR MODIFY IT UNDER THE TERMS OF THE GNU GENERAL PUBLIC LICENSE AS PUBLISHED BY THE FREE SOFTWARE FOUNDATION, EITHER VERSION 3 OF THE LICENSE, OR (AT YOUR OPTION) ANY LATER VERSION. WNS IS DISTRIBUTED IN THE HOPE THAT IT WILL BE USEFUL, BUT WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SEE THE GNU GENERAL PUBLIC LICENSE FOR MORE DETAILS. YOU SHOULD HAVE RECEIVED A COPY OF THE GNU GENERAL PUBLIC LICENSE ALONG WITH WNS. IF NOT, SEE AT HTTP://WWW.GNU.ORG/LICENSES THERE IS NO WARRANTY FOR WNS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE WNS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. LIMITATION OF LIABILITY. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE WNS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF WNS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. DISCLAIMER ON THE DOCUMENTATION AND WEB SITE HTTP://WWW:WNSDOCK.XYZ OF WNS. IN NO WAY CAN ANY RIGHTS BE DERIVED FROM, OR CLAIMS MADE, WITH REGARD TO THE CONTENT OF THIS WEBSITE AND THE DOCUMENTATION THEREIN. ALTHOUGH THE GREATEST POSSIBLE CARE HAS BEEN TAKEN WITH THE COMPILATION OF THE CONTENT OF THIS WEBSITE AND DOCUMENTATION, IT IS POSSIBLE THAT CERTAIN INFORMATION MAY (AFTER A WHILE) BE OUT-OF-DATE OR (NO LONGER) BE CORRECT. WE ARE NOT RESPONSIBLE FOR EVENTUAL DAMAGES ARISING FROM THE USE OF INFORMATION FROM THIS SITE. WE HEREBY REJECT ALL RESPONSIBILITY FOR DAMAGES AS A RESULT OF THE USE OF THIS INFORMATION OR INFORMATION TO WHICH LINKS REFER ON THIS SITE (THESE SITES). THE INFORMATION ON THIS SITE MAY BE CHANGED WITHOUT PRIOR WARNING. WE DO NOT GIVE GUARANTEES WITH REGARD TO THE NATURE AND THE CONTENT OF THIS SITE. ALL RESPONSIBILITY FOR POSSIBLE DAMAGES DUE TO ACCESS TO AND USE OF THE SITE IS EXPLICITLY REJECTED BY US. WE DO NOT ACCEPT ANY RESPONSIBILITY WITH REGARD TO THE CONTENT, ADVERTISEMENTS, PRODUCTS, OR OTHER ISSUES ON SUCH SITES OR SOURCES OR AVAILABILITY. WE ARE NOT RESPONSIBLE FOR ANY KIND OF DAMAGE OR LOSS CAUSED BY OR IN CONNECTION WITH THE USE OF, OR BY RELYING ON THE CONTENT, PRODUCTS OR SERVICES OFFERED ON SUCH SITES OR SOURCES. COPYRIGHT (C) MONIKA BALINT AND CSABA HETENYI, 2017, 2018.